IMS Learning Design : notational language, modelling language or design language ?

Jean-Philippe PERNIN (*, **) 
Anne LEJEUNE (**) 

(*) Institut national de recherche pédagogique ERTé e-Praxis 
(**) Laboratoire CLIPS-IMAG, Grenoble
Plan

• Analysis based on a software engineering approach centred on users.
  – IMS LD specification ambiguity
  – What are users and needs?
  – What is precisely object of study and lifecycle
  – What tools to develop and for whom
  – What we have to standardize or not

• Propose solutions
  – Scenarios, lifecycle et facets
  – Suggesting activities for French community
IMS LD : a double ambiguity

• Ambiguity of intentions
  – Notational system
  – Educational modelling language
  – Learning Design

• Ambiguity of use contexts
  – by who, for whom, when ?
    • Industrialised context, handicraft development ?
  – Which contexts
    • Distant, face to face, blended learning

• *Ambiguities probably caused by abusive interpretations*
Concevoir, modéliser, noter

- Three different activities gathered in a same framework : a potential source of ambiguity
  
  - **Design** : imagine a solution and share it within a community with same references and goals
  
  - **Modelling** : express the most precisely the designed solution, with a formalism accessible and leading on a specific paradigm
  
  - **Notating** : express a modelised solution with a common language for facilitating its interchange and its execution in various contexts
Parallèle avec les langages de programmation

Environments and design languages (ex.: Powerpoint)

Programming language

Assembly and machine languages

1er avril 2005   J.P.Pernin et A. Lejeune
Langages et scenarios d'apprentissage

Specific design languages and environments

Modelling language

Notational system

IMS - Learning Design ?

Teachers and trainers

Specialists of modelling

LMS
IMS LD as a notational system (1)

• For who ?
  – For the computers : insure automation and interoperability
  – For which users ?
    • "programmers" able to write and read in native language ?

• Main criteria :
  – Interoperability
  – Completeness
  – Ability to be executed, efficiency, concision
  – Readability ?
IMS LD as a notational system (2)

• Pros :
  – Interoperability : OK (XML)

• Cons :
  – Completeness : To validate (in progress)
    • Presence of metaphor limiting expressiveness ?
    • Taking account of production ?
  – Ability to be executed within a LMS
    • To validate (in progress)
IMS LD as a modelling language (1)

• For who?
  – Modelization specialist, from a semi formal description (UML diagrams, narratives, etc.)

• Main criteria:
  – Readability
  – Pedagogical neutrality
  – Framework accessibility
  – Matching between the used metaphors and the situations to model
IMS LD as a modelling language (2)

• Pros:
  – Conceptual framework powerful (Roles, Activities, Environment),
    • …but is it the only one?
  – Powerful of theatrical metaphor
    • … but isn't it a limitation?

• Cons:
  – Heterogeneous and complex to manipulate levels of abstraction
    • Levels A, B et C: What is the exact semantic?
    • Properties and conditions
      – what level: modelling or notation?
IMS LD as a design language (1)

A suggested process: 3 steps reflecting an industrialized context of development

Design:
Scenario informal (textual)

Narrative

Modelling: semi-formal step

UML

Notation: formal step

IMS LD
IMS LD as a design language (2)

• For who?
  – Teachers and trainers belonging to specific communities of practice

• Criteria:
  – Accessibility, usability
  – Reusability

• Pros and cons:
  – IMS LD is not specified for that purpose

• Enjeu
  – Get available model to teachers (Goals of UNFOLD, 2005)
IMS LD : our point of view

- A real improvement for describing situations centred on activities
- Unavoidable base for elaboration of a notational system
- Unsatisfying modelling language
- Not adapted to design by non-specialists
Organisation des différents niveaux

Communities of practice

Environment or design language

Paradigm A

Modelling language

Modeller

Environment or design language

Modelling language

Environment or design language

Environment or design language

Paradigm B

Standardised notational language

LMS

1er avril 2005

J.P.Pernin et A. Lejeune
Synthesis: differentiate 3 levels

• Learning Notational Languages
  – The effort of standardisation must be focused on that feature
  – Must IMS LD be simplified, enriched?

• Learning Modelling Languages?
  – Necessity of standardization?
    • Diversity of approaches (different paradigms)
    • Reach a consensus, if possible

• Learning Design Authoring Languages?
  – Detect current design uses in order to propose adapted tools to users
Part 2: propositions and current works

- Glossary and lifecycle of scenarios
- Propose improvements of conceptual models: the different facets of a scenario
- Propose adapted tools
Learning scenario definition

[Pernin&Lejeune 2004]

– A learning scenario represents a description, made a priori or a posteriori, of the progress of a learning situation at a given level, or learning unit, whose goal is to ensure the appropriation of a precise set of knowledge. A scenario describes roles, activities and also knowledge resources, tools and services necessary or resulting for each activity.
The facets of a scenario

• Every learning situation can be
  – prescribed activities
  – Observation of
  – Capitalisation
  – Regulation

• Scenario modelling language
  – Must be able to express those facets in a abstract way (without programming formalisms such as properties and conditions)
Scenario Lifecycle

1. Design
   - Create
   - Reuse
   - Adapter

2. Contextualization
   - Distributing roles
   - Planning activities
   - Affecting resources
   - Instrument
   - Refine

3. Execution
   - Prescribe
   - Observe
   - Capitalise
   - Regulate

4. Use feedbacks
   - Evaluate
   - Decontextualise
   - Catalog

adapted scenario
contextualised scenario
descriptive scenario
scenario pattern

Reusing

1er avril 2005 J.P.Pernin et A. Lejeune
## Taxonomy of scenarios

<table>
<thead>
<tr>
<th>Variable criteria</th>
<th>informal</th>
<th>formalized</th>
<th>automatizable</th>
</tr>
</thead>
<tbody>
<tr>
<td>degree of formalization</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>degree of abstraction</td>
<td>abstract</td>
<td>concrete</td>
<td></td>
</tr>
<tr>
<td>Constant criteria</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>finality</td>
<td>predictive</td>
<td>descriptive</td>
<td></td>
</tr>
<tr>
<td>granularity</td>
<td>activity</td>
<td>sequence</td>
<td>structuration</td>
</tr>
<tr>
<td>degree of personalization</td>
<td>generic</td>
<td>adaptative</td>
<td></td>
</tr>
<tr>
<td>Degree of constraint</td>
<td>constrained</td>
<td>open</td>
<td>adaptable</td>
</tr>
</tbody>
</table>
What tools for scenarios

- Dynamic adaptation functionalities
- Deployment and execution functionalities
- Assistants for contextualization
- Assistants for design of abstract scenarios

Indexing and searching tools of scenario patterns

Prescription, Observation, Control

Effective scenario
adapted scenario
contextualized scenario
abstract scenario
scenario pattern

Environment

Activities

Results

Rôles

Abstract productions
abstract Tracks

concrete resources
concrete Services

Localised
Localised

Distributed rôles

Plan

Localised

tracks localised

Localised

activities planed

Productions localised

Abstract resources
Abstract Services

situated resources
situated Services

Rôles distribute

Mediatise

Instrument
Possible contribution

• Suggestions
  – Separating formally
    • Learning Notational Language
    • Learning Modelling Language
  – Standardised solutions for deployment
    • projection Languages from LD towards LMS?

• Take account of :
  – Users needs and practices
  – Abilities of implementation of LMS